UNITED STATES PATENT AND TRADEMARK OFFICE

Fax Cover Sheet

Date: 20 Apr 2010

To: Peter Martine	**From:** Erika Kretzmer
Application/Control Number: 10/550,266	**Art Unit:** 2192
Fax No.:	**Phone No.:** (571) 270-5554
Voice No.: 408-749-6900	**Return Fax No.:** (571) 273-8300
Re: App. 10/550,266; Attorney Docket WACHP008	**CC:** SPE Tuan Dam

☐ **Urgent**   ☒ **For Review**   ☒ **For Comment**   ☒ **For Reply**   ☐ **Per Your Request**

Comments:
See next page. Proposed claim 1 amendments attached,

Number of pages 3 including this page

Examiners want to confirm with Applicant that "an operating state" includes a value written to the common heap. Such interpretation of claim 1 was presented in the first rejection.

See proposed Examiner's Amemendments to claim 1 on the following page. Examiners propose making the same amendments to claims 10 and 21, and cancelling claims 7, 18, and 23.

Suggest adding "for each execution of the program" to claim 1, to get around art where the program is executed once, then checked against a signature or trace on each subsequent execution (such as "Oblivious Hashing" (Chen et al., 2002), not yet cited). Supported by paragraph 32 of specification.

Suggest removing "wherein write operations to the common heap are only performed by one of the first and second virtual machines" because performing write operations by the second VM is not supported in the specification.

Suggest incorporating the limitation "wherein, when an instruction of the program that contains a write operation to the common heap is being executed, a write operation is performed only by the first virtual machine" from claim 7, because this appears to have the same function as the removed limitation.

These amendments would put the claims in condition for a Notice of Allowance to be issued. Please review and reply.

Claim 1 (Currently Amended): A method for the controlled execution of a program, the program being intended for a virtual machine, on a portable data carrier, wherein

the data carrier has a processor which executes at least a first and a second virtual machine <u>for each execution of the program</u>, (paragraph 32)

the program is executed both by the first and by the second virtual machine,

the first and the second virtual machine both access a common heap in a non-volatile memory of the data carrier, [wherein write operations to the common heap are only performed by one of the first and second virtual machines,] <u>wherein, when an instruction of the program that contains a write operation to the common heap is being executed, a write operation is performed only by the first virtual machine,</u> (Claim 7)

an operating state of the first virtual machine and an operating state of the second virtual machine are checked during execution of the program for correspondence,

and

execution of the program is aborted if a difference is found between the operating state of the first virtual machine and the operating state of the second virtual machine.